

Real-Time Human Detection and Tracking from Mobile Vehicles

Wael Abd-Almageed Mohamed Hussein Mohamed Abdelkader Larry Davis
Institute for Advanced Computer Studies
University of Maryland at College Park
wamageed, mhusein, mdfarouk, lsd@umiacs.umd.edu

Abstract—This paper presents a real-time system for detecting and tracking humans from mobile vehicles. The system integrates human detection and tracking algorithms and employs depth information obtained from a stereo vision system. Depth information is used to limit the search space of the detector. We also present a simpler and faster variant of the popular Adaboost human detector. Experimental results demonstrate that integrating depth information vastly reduces the false detection rate compared with detection without stereo. Also, we show that integrating stereo enables the detection and tracking subsystem to operate in real-time.

I. INTRODUCTION

Human detection and tracking have been an extremely active area over the past decade. The importance of this area arises from its numerous applications such as smart vehicles [13][15], military applications [14] and security systems [4]. However, it has been shown that detecting humans from a single image while maintaining a low false detection rate is a very difficult problem. The difficulty is mainly due to the lack of distinguishing visual features that characterize human appearance. The literature [13][8][19] shows that most effective features are based on the gradient (or edge map) of the image of interest.

This paper is organized as follows. Section II briefly reviews the related work in the area. The system overview and technical details are discussed in Section III. Finally, the experimental results are demonstrated in IV. Section V concludes the paper.

II. RELATED WORK

We can broadly classify the work in human detection into two main categories. The first category is detecting human figures in still images, or what is called shape-based human detectors, such as the work of Gavrila and Philomin [13], Gavrila and Munder [12], Dalal and Triggs[8] and Zhu et al. [19]. These methods extract gradient (or edge) features and either match these features against human templates or use a binary classifier to decide whether or not the extracted features are from a human. The second category is the so-called motion-based human detectors, such as the work of Cutler and Davis [7], Ran et al. [17] and Abd-Almageed et al. [2]. This class of algorithms depends on tracking an object of interest for a short period of time (typically 2-4 seconds) and analyzing the motion pattern of the object. A tracked object

is classified as a human if it exhibits a twin-pendulum-like periodic motion.

Shape-based detectors suffer from two main drawbacks; high false detection rate and slow performance, since the entire image has to be scanned (at multiple scales) to find human figures. The main advantage of this class of algorithms, however, is that they do not need to be initialized. On the other hand, the main advantage of the motion-based methods is the low false detection rate. However, an object of interest must first be detected in order to employ motion-based methods.

In [14] we have shown that an effective strategy to reduce the overall false detection rate of a human detection and tracking system is to combine both shape-based and motion-based methods. Shape-based methods are first used to detect potential human objects. An object tracker is then used to track the object for a sufficient period of time. Finally, the motion of the tracked object is analyzed to verify if it resembles the motion of a human. Figure 1 shows the result of applying the shape-based detector only (red boxes) and the result of a combined shape/motion detector (green boxes). The high false detection rate of the shape detector is due to the sensitivity of the detector to highly cluttered areas. After tracking such false detections for a short period of time, the motion analyzer can easily determine that these false alarms do not exhibit a human-like motion. Gavrila and Philomin in [13] introduced a fast human detection algorithm based on the distance transform (chamfer distance) [6]. They collected a database of silhouette images of humans in different poses. During the training phase, K-means was used to cluster the silhouette database based on the pairwise distance, into a number of clusters. K-means is then repeatedly applied to each cluster to further cluster it into a number of sub-clusters. The process is repeated yielding a hierarchy of human silhouettes. To detect if a human exists in a given image (of the same size as the training images), edges are extracted from the test image and the distance is computed between the edge map and silhouettes in the hierarchy. A human is detected if the computed distance is smaller than a pre-specified threshold across all levels of the hierarchy. The main advantage of this algorithm is speed. However, the algorithm is highly sensitive to image clutter and noise. Recently, Gavrila and Munder [12] integrated the detecting algorithm with stereo vision in order to lower the number of false alarms. Dalal and Triggs in [8] introduced a learning-based algorithm to detect humans from a single

This work was funded, in part, by Army Research Laboratory's Robotics Collaborative Technology Alliance program; contract number DAAD 19-012-0012 ARL-CTA-DJH.

images. The image is divided into 16×16 rectangular neighborhoods and a feature vector called the Histogram of Oriented Gradients (HoG) is computed for each neighborhood. The HoG represents the probability distribution of gradient orientation (quantized into a pre-defined number of histogram bins) over a specific neighborhood. All HoGs from all image neighborhoods are concatenated to form a larger feature vector describing the image of interest. A Support Vector Machine (SVM) is used to classify if the given sub-image contains or does not contain a human. In general, this method has a lower false alarm rate than that of Gavrilu and Philomin [13]. However, to check for humans at different scales, computing the HoG feature vector and using the SVM to classify it becomes computationally expensive; and therefore the detector is relatively very slow. It was reported in [8] that the algorithm runs at 1 frame per second if 800 detection windows from a 320×240 image are selected and analyzed. In [19], Zhu et al. used a cascaded Adaboost algorithm [10][18] to rapidly detect humans in static images using the HoG feature vector. In order to improve the overall performance, they employed the concept of integral images [18] to compute the feature vector. A Support Vector Machine classifier was used as the weak classifier of the Adaboost algorithm. Two main problems remain not addressed. First, to detect humans at a given scale, the entire image has to be blindly scanned pixel by pixel. The second problem is that this process must be repeated for an arbitrary number of scales in order to find all humans at different distances from the moving vehicle. These two problems increase the computational requirements and false detections of these algorithms.

III. SYSTEM DETAILS

The system we developed in [14] had a major weakness. To find initial detections and because of the monocular nature of the system, the entire image needed to be searched many times to find humans at different scales. Consequently, this significantly increased both detection time and the false detection rate. Moreover, since we had no prior knowledge regarding the scales of humans in the scene, a large number of scales was used, which again increased the false detection rate.

To solve this problem, as we discussed in Section II, all of the initial detections (generated by the shape-based detector) were tracked for a short period of time. A motion-based detection algorithm was then applied to the tracked object to analyze its motion. An initial detection is declared a human if it exhibits human-like motion.

The following sub-sections briefly discuss the different modules that comprise our detection and tracking system. In Section III-C we present the architecture and implementation details of the system.

A. Fast Image Search using Depth Information

We adopt a two-stage approach for using depth information. The first stage is the estimation of image pixels that correspond to the ground plane. Labayrade et al. [15]

introduced a simple and fast algorithm for ground plane estimation, known as v-Disparity algorithm. Briefly, the v-Disparity algorithm computes histograms of disparity values for each image row and searches for the peaks of these histograms. The bin at which the histogram of a specific row peaks is assumed to correspond to the disparity of the ground plane at this row. Figure 1 shows an example of applying the v-Disparity algorithm. Figure 1.b shows the disparity map of the image in 1.a. The result of applying v-Disparity is illustrated in Figure 1.c, where the red area represents the estimated ground plane area, the green area represents non-ground plane areas and the blue areas represents area where the disparity is not defined.

The estimation of the ground plane significantly limits the search space of the detector for two reasons. Firstly, the detector is only used at non-ground plane areas, which eliminates a large part of the image. Secondly, only areas where a potential detection will touch the ground plane are considered. In other words, we only search for humans standing (or walking or running) on the ground plane rather than flying humans.

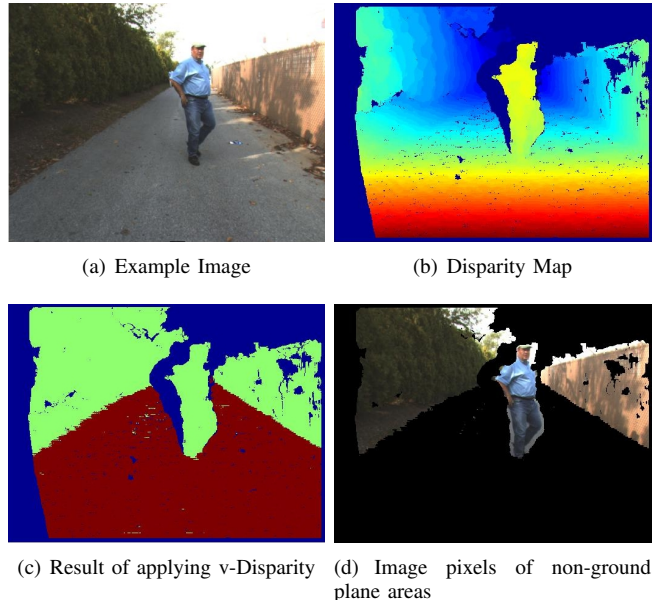


Fig. 1. Result of using the v-Disparity algorithm. Red, green and blue areas of Figure 1.c represent ground plane, non-ground plane and undefined disparity areas, respectively

The second stage of analysis of the depth information finds the peaks of the density of the disparity information. We use the mean shift procedure which was first presented in [11]. Let

$$\mathbf{X} = \{x_i; i = 1, \dots, N\} \quad (1)$$

be the set of disparity values x_i of the non-ground plane area where the disparity is defined and N is the number of set elements. The probability density of \mathbf{X} can be non-parametrically estimated using the kernel density estimator

of Equation

$$p(x) = \frac{1}{n} \sum_i \frac{1}{h_N} \Phi\left(\frac{x - x_i}{h_N}\right) \quad (2)$$

where Φ is kernel function (Gaussian kernel in our case) and h_N is the bandwidth. Taking the derivative of Equation 2 gives an estimate of the gradient of the probability density of x_i , as shown in Equation 3

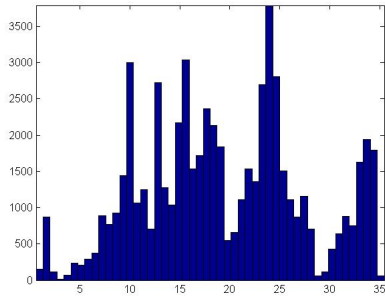
$$\nabla p(x) = \frac{-1}{n h_N} \sum_i \Phi'\left(\frac{x - x_i}{h_N}\right) \frac{\sum_i x_i \Phi'(x - x_i)}{\sum_i \Phi'(x - x_i)} - x \quad (3)$$

where

$$m_x = \frac{\sum_i x_i \Phi'(x - x_i)}{\sum_i \Phi'(x - x_i)} - x \quad (4)$$

is called the mean shift vector. By iteratively computing the mean shift vector m_x and shifting the kernel function $\Phi(\cdot)$, the procedure converges to the local maxima (i.e. modes) of the density of x_i , yielding a set of disparity values $\mathbf{X}_c = \{x_j; j = 1, \dots, M\}$ where $M \ll N$. The values of the set \mathbf{X} are clustered into M clusters based on the distance $|x_i - x_j|$.

Figure 2.a shows the histogram of the disparity values of the non-ground plane area (where disparity is defined) of Figure 1.b. The result of clustering the disparity values to the modes of the density obtained by the mean shift algorithm is shown in Figure 2.b.



(a) Histogram of disparity values of Figure 1.b



(b) Clustering the disparity value using the mean shift algorithm

Fig. 2. Results of clustering the disparity map using the mean shift algorithm

By clustering the disparity values, we determine the search scale that should be used for each window of the image of interest; rather than repeatedly applying the human detector for an arbitrary large number of scales. This, in fact, has two

impacts on the detection results; reducing the false detection rate and speeding up the detection process.

B. Fisher Linear Discriminant Human Detector

We implemented a fast and accurate human detector based on the work of Zhu et al. [19], which combined the Histogram of Oriented Gradients descriptors [8] and a cascade of boosted feature selectors, introduced in [18].

In [19], the feature selectors select the best block out of randomly selected 5% of the total number of available blocks. The total number of available blocks was reported to be 5031. That means only 252 blocks are searched by the feature selectors. In our implementation, we deployed a coarser method for image window division that gave us a total of 2748 different blocks instead of 5031. However, instead of searching a random sample therefrom, our feature selectors search the entire domain of available blocks to find the best block. Despite the one order of magnitude increase in the number of blocks searched in our implementation, the training time is much shorter. That is due to a simplification in the classifier, as explained next. Our coarse image window division works as follows: a search window of size 64×128 is divided into block of sizes from 12×12 to 64×128 , with increment of 4 in each dimension. The aspect ratios of blocks are either 1 : 1, 1 : 2, or 2 : 1. For the spatial step between blocks, we use step of 4 for dimension sizes of 12 or 16, step of 6 for dimension sizes of 20 or 24, and step size of 8 otherwise.

Zhu et al. [19] used an SVM with a linear kernel as the weak classifier used by feature selectors. To speed up the training process and to enable searching the entire domain of available blocks, we used a simpler linear classifier based on the Fisher Linear Discriminant [9] as described by Equation 5

$$\vec{w} = (\Sigma_{y=0} + \Sigma_{y=1})^{-1}(\vec{\mu}_{y=1} - \vec{\mu}_{y=0}) \quad (5)$$

where $\Sigma_{y=0}$ and $\mu_{y=0}$ are the covariance matrix and mean vector, respectively, of feature vectors (HoG) of the human class and $\Sigma_{y=1}$ and $\mu_{y=1}$ are the covariance matrix and mean vector, respectively, of feature vectors (HoG) of the human-free examples. Computing the linear classifier \vec{w} from the training data at each stage of the cascade is computationally inexpensive compared to the computational complexity of training SVMs. However, by boosting a number of \vec{w} , we can achieve classification accuracy similar to that of using SVMs, which is the main reason of using Adaboost [10].

In training the cascade, we used the positive training data in the INRIA person dataset [8], which consists of 2416 pedestrian images. In order to build a large repository of negative examples (approximately 6000 images), we used the INRIA's negative images, human-free images from [16][1] and a number of images from the Internet. During the training process, we maintained the number of negative examples in training each layer of the cascade at 3020. Specifically, to train a new layer of the cascade, random windows from our negative image repository are selected and only the windows that are classified as positive by the

previously trained layers are kept. This process is repeated until the required number of negative image windows for the new layer is reached. In our best classifier, the target false detection rate per layer was set to 0.75, the target detection rate per layer was set to 0.998, and the over all target false alarm rate was set to 10^{-4} . The resulting classifier consisted of 28 layers, with 621 weak classifiers in total.

C. System Architecture

The system mainly consists of two concurrently running threads for detection and tracking, as illustrated in Figure 3. The detection algorithm is applied every T frames. Detection results are added to a queue of detections shared between the two threads.

The tracking thread is continuously running until receiving an end signal from the detection thread meaning no more frames are being received. We use the algorithm we have previously developed for tracking human objects [5]. The tracker models the appearance of the object using mixtures of Gaussians [3] and uses particle filters to estimate the state of the object being tracked.

IV. RESULTS

We tested our trained classifiers on the test images of the INRIA person dataset. A Detection Error Tradeoff (DET) curve depicting False Positives Per Window (FPPW) vs Miss Rate, was plotted for our best trained classifier and compared to the one reported in [19], for the case of L1 normalization (since we only use L1 normalization.) We found that the two curves are similar to one another with a slight lead to the one in [19]. DET curve for our best trained classifier is shown in Figure 4. For comparison, the reader is referred to [19].

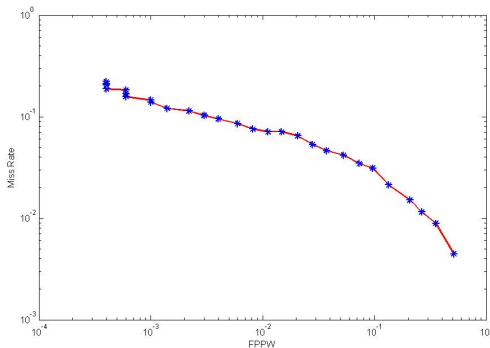


Fig. 4. DET curve for our best trained classifiers.

Figure 5 shows the results of applying the detection algorithm of [19] without using depth information (left column) and the result of applying our Fisher classifier with using depth information (right column). The detection rate was set to $T = 60$. The figure illustrates that using depth vastly reduces the false detections. The execution time of [19] was 0.83 frames per second. The execution time for our algorithm was 3.5 frame per second; a 4.2 fold improvement on a 512×384 image.

The results of tracking a running pedestrian are shown in Figure 6. The person being tracked changes direction of motion as well as speed a number of times through out the sequence. Our tracker runs at 30 frames per second on images of size 320×480 . We use full resolution images for detection and then subsample the image to a lower resolution in order to run the tracker in real time.

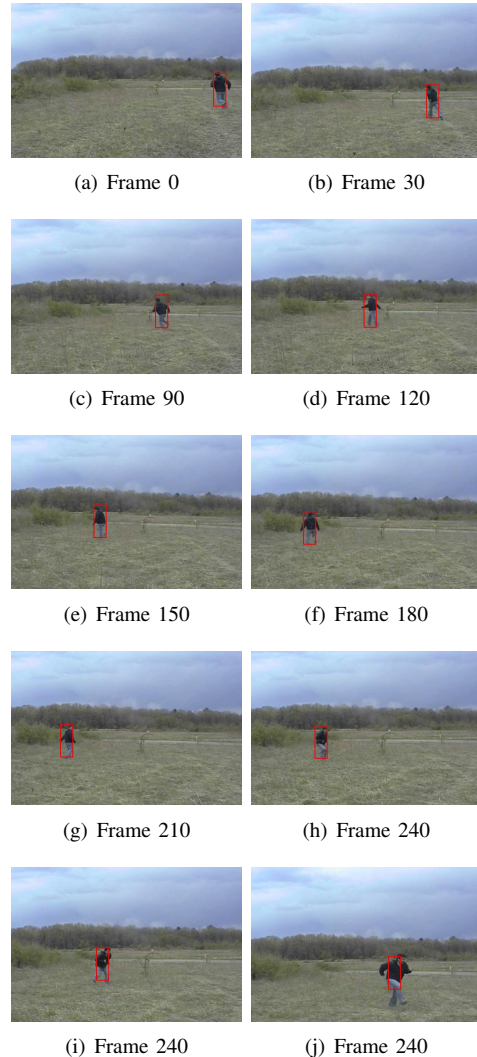


Fig. 6. Human tracking results using the tracking algorithm of [5]

V. CONCLUSIONS AND FUTURE WORK

We presented a system for detecting and tracking humans in real-time. There are three main contributions for the paper. We show how the false detection rate can be significantly reduced by integrating depth information into human detection algorithms. Also, by integrating depth information, we are able to estimate the appropriate search scales, which speeds up the detection process.

Moreover, we have presented a variant of the cascaded Adaboost detector of [19]. Our detector uses a simpler weak classifier based on Fisher Linear Discriminants. The low complexity of our classifier speeds both training and testing

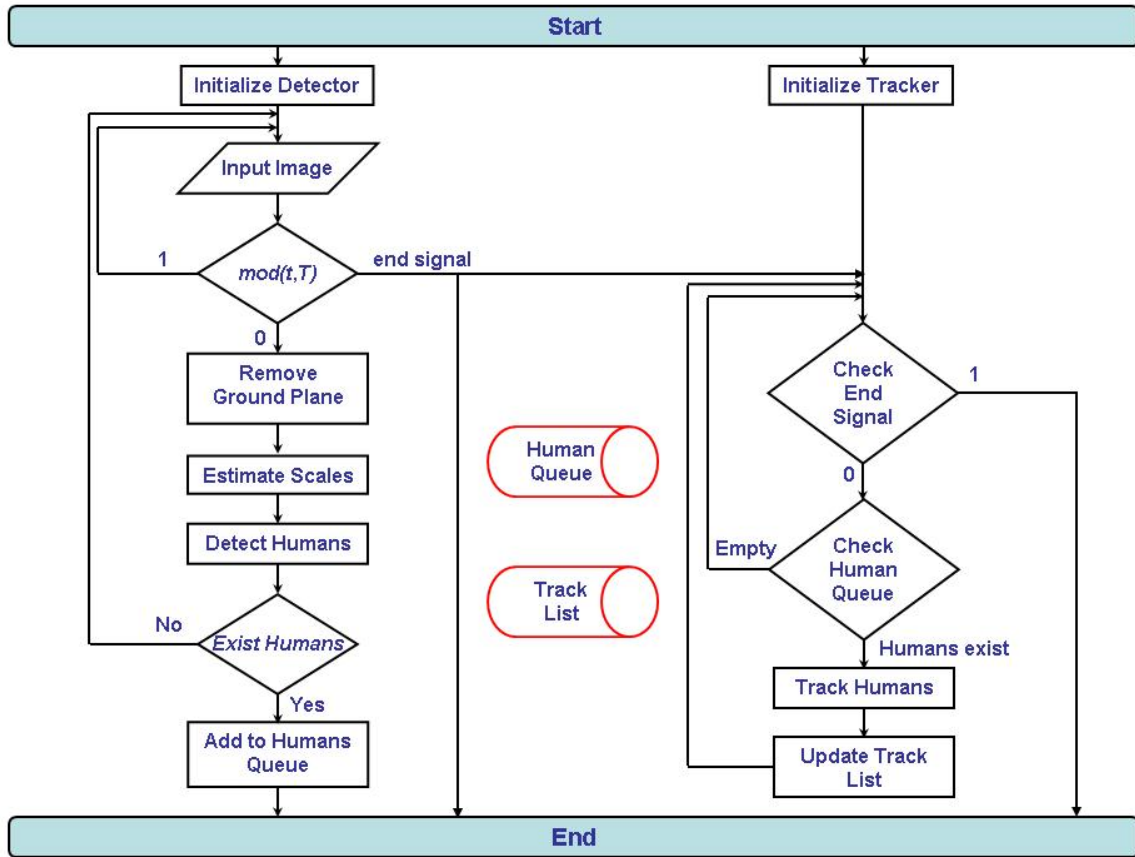


Fig. 3. System Architecture. Two threads for detection and tracking are running in parallel. The detector runs at a user-specified rate T . The two threads communicate through a termination signal sent from the detector thread to the tracker thread, a human detection queue and a track list

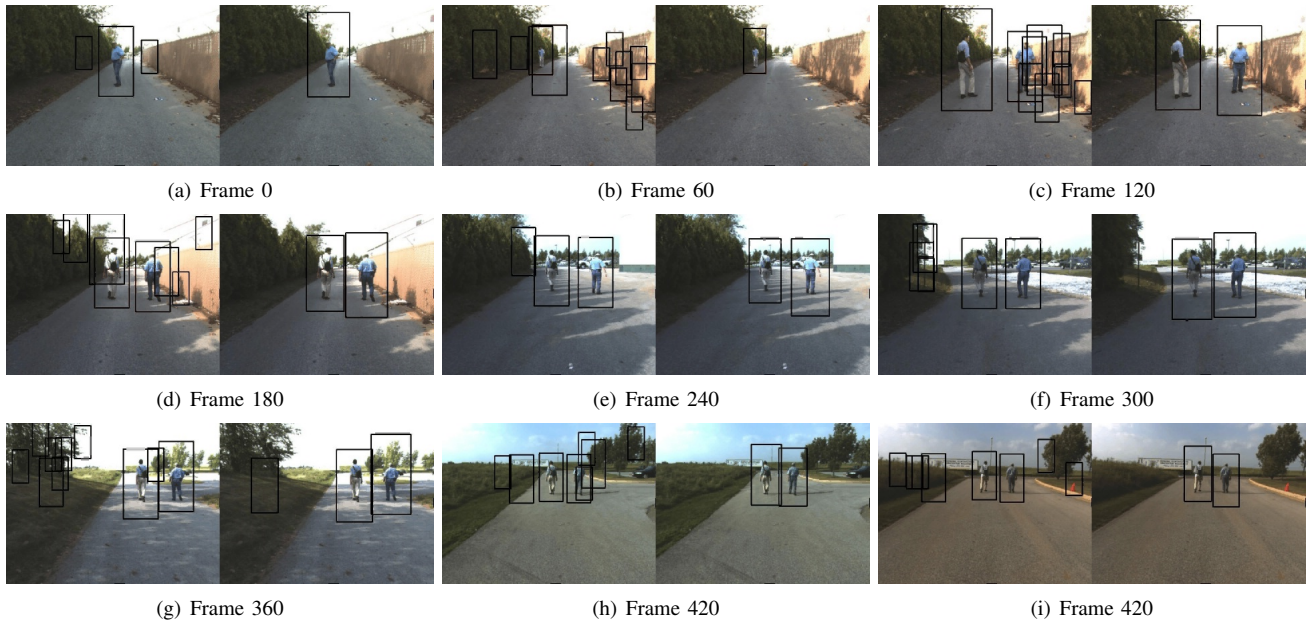


Fig. 5. Detection results. Left column shows the detection results of the algorithm of Zhu et al. [19]. Right column shows the detection results of our stereo-assisted detection algorithm illustrated in Section III. Our detector generates far less false detection because of using depth information to guide the search. Detection rate is set at $T = 60$ (i.e. detector runs every two seconds)

phases while maintaining similar detection performance to that of [19]. Finally, a multi-threaded human detection and tracking system was presented.

We are currently investigating two system improvements. First, rather than periodically estimating the search scales, we are developing a system component which automatically determines when to update the scale. Secondly, we are developing a variant of the cascaded Adaboost detector incorporating depth information.

ACKNOWLEDGMENT

We would like to thank Dr. Andrew Howard and Dr. Arturo Ranking from Jet Propulsion Laboratory (JPL) for providing the stereo data. We also would like to thank General Dynamics Robotic Systems for supporting the data collection.

REFERENCES

- [1] "Caltech image dataset." [Online]. Available: www.vision.caltech.edu/html-files/archive.html
- [2] W. Abd-Almageed, B. Burns, and L. Davis, "Identifying and segmenting human motion for mobile robot navigation using alignment errors," in *IEEE International Conference on Advanced Robotics*, 2005.
- [3] W. Abd-Almageed and L. Davis, "Density estimation using mixtures of mixtures of gaussians," in *European Conference on Computer Vision*, 2006.
- [4] —, "Robust appearance modeling for pedestrian and vehicle tracking," in *Lecture Notes in Computer Science 4122, Proc. of Classification of Events, Activities and Relationships (CLEAR) Workshop*, R. Stiefelhagen and J. Garofolo, Eds., 2007, pp. 209–215.
- [5] W. Abd-Almageed, M. Hussein, and L. Davis, "Tracking articulating objects from ground vehicles using mixtures of mixtures," in *IEEE International Conference on Intelligent Robots and Systems*, 2006.
- [6] G. Borgefors, "Distance transformations in digital images," *Computer Vision, Graphics and Image Processing*, vol. 34, no. 3, 1986.
- [7] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.
- [8] N. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.
- [9] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. II, pp. 179–188, 1936.
- [10] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, 1997.
- [11] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Information Theory*, vol. 21, pp. 32–40, 1975.
- [12] D. M. Gavrilu and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, June 2007.
- [13] D. M. Gavrilu and V. Philomin, "Real-time object detection for smart vehicles," in *IEEE International Conference on Computer Vision*, 1999.
- [14] M. Hussein, W. Abd-Almageed, Y. Ran, and L. Davis, "A real-time system for human detection, tracking and verification from a moving camera," in *IEEE International Conference on Computer Vision Systems*, 2006.
- [15] R. Labayrade, D. Aubert, and J. P. Tarel, "Real-time obstacle detection on non flat road geometry through v-disparity representation," in *IEEE Intelligent Vehicles Symposium*, 2002.
- [16] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
- [17] Y. Ran, I. Weiss, Q. Zhen, L. Davis, and W. Abd-Almageed, "Pedestrian classification from moving platforms using cyclic motion pattern," in *IEEE International Conference on Image Processing*, 2005.
- [18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.
- [19] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng, "Fast human detection using a cascade of histogram of oriented gradients," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.